京东注册场景测试2

一、背景

鉴于京东反馈的数据标的，对京东数据进行二次测试。此次测试，主要采用三种有监督的机器学习算法，分别是极端梯度上升算法（xgboost）、梯度提升决策树算法（GBDT）、随机森林算法（RandForest），进行建模并预测数据是否存在欺诈风险。

二、打标数据及无监督测试结果对比

京东打标数据：分数>0为黑样本，分数<=0 为白样本； 其中去重之后黑样本85861条,白样本12443条。

打标数据和第一次测试数据结果比对：

Web端黑样本14007条，总的准确率35.73%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | all |
| 1 | 2188 | 4430 | 1241 | 1319 | 5005 |
| 0 | 11819 | 9577 | 12766 | 12688 | 9002 |

App端黑样本71854条,总的准确率为6.37%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | all |
| 1 | 1552 | 3644 | 197 | 231 | 4574 |
| 0 | 70302 | 68210 | 71657 | 71623 | 67280 |

黑样本总共85861条，总的准确率为11.16%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | all |
| 1 | 3740 | 8074 | 1438 | 1550 | 9579 |
| 0 | 82121 | 77787 | 84423 | 84311 | 76282 |

web端白样本1068条，总的准确率为88.30%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | is\_risk |
| 1 | 106 | 30 | 3 | 3 | 125 |
| 0 | 962 | 1038 | 1065 | 1065 | 943 |

APP端白样本11375条，总的准确率为87.84%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | is\_risk |
| 1 | 213 | 1267 | 13 | 28 | 1383 |
| 0 | 11162 | 10108 | 11362 | 11347 | 9992 |

白样本总量12443条，总的准确率为87.88%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试结果 | GMM | rule | iforest | graph | is\_risk |
| 1 | 319 | 1297 | 16 | 31 | 1508 |
| 0 | 12124 | 11146 | 12427 | 12412 | 10935 |

结论：通过反馈的数据对比，可以看出：

* 黑样本Web端的准确率要高于App端，但准确率均不高
* 白样本准确率均比较高

主要原因有两点：

1. 因为京东传入的数据有修改，时间与事件发生不一致；
2. 数据分布不一致：第一次测试采用无监督的方法，是假设大多数是正常样本，少数是黑样本；但通过沟通，发现黑样本比例在50%以上，因此无监督找异常点的方法并不合适。

因此，鉴于打标数据，重新采取有监督的方法进行建模预测。

三、数据源分析

1、数据源描述

京东反馈打标数据：共100001条记录，包含两个字段，账号、分数。分数>0为黑样本，分数<=0 为白样本； 其中去重且与测试数据能关联的黑样本85861条,白样本12443条。

|  |  |  |  |
| --- | --- | --- | --- |
| 渠道 | 黑样本 | 白样本 | 总计 |
| Web | 14007 | 1068 | 15075 |
| app | 71854 | 11375 | 83229 |
| 总计 | 85861 | 12443 | 98304 |

2、特征构建：

以用户注册事件作为主维度，从横向和纵向两个维度构建特征集合。

**横向特征**：构建设备、IP等字段关联的全局特征。（缺失手机号，邮箱等数据）。

**纵向特征**：构建事件的时序特征，以注册事件的时间为基准，向前回溯一段时间，或者N个事件的特征。

构造纵向特征共计36个，横向关联特征流量29个。

|  |  |  |
| --- | --- | --- |
| 特征英文名 | 特征中文名 | 是否时间特征 |
| v\_dev\_1min\_cnt | 设备1分钟内出现次数 | 是 |
| v\_dev\_10min\_cnt | 设备10分钟内出现次数 | 是 |
| v\_dev\_1hour\_cnt | 设备1小时内出现次数 | 是 |
| v\_dev\_1day\_cnt | 设备1天内出现次数 | 是 |
| v\_dev\_1min\_account\_cnt | 设备1分钟关联账户数 | 是 |
| v\_dev\_10min\_account\_cnt | 设备10分钟关联账户数 | 是 |
| v\_dev\_1hour\_account\_cnt | 设备1小时关联账户数 | 是 |
| v\_dev\_1day\_account\_cnt | 设备1天关联账户数 | 是 |
| v\_dev\_1min\_ip\_cnt | 设备1分钟关联ip数 | 是 |
| v\_dev\_10min\_ip\_cnt | 设备10分钟关联ip数 | 是 |
| v\_dev\_1hour\_ip\_cnt | 设备1小时关联ip数 | 是 |
| v\_dev\_1day\_ip\_cnt | 设备1天关联ip数 | 是 |
| v\_ip\_1min\_cnt | IP1分钟内出现次数 | 是 |
| v\_ip\_10min\_cnt | IP10分钟内出现次数 | 是 |
| v\_ip\_1hour\_cnt | IP1小时内出现次数 | 是 |
| v\_ip\_1day\_cnt | IP1天内出现次数 | 是 |
| v\_ip\_1min\_account\_cnt | IP1分钟内关联账户数 | 是 |
| v\_ip\_10min\_account\_cnt | IP10分钟内关联账户数 | 是 |
| v\_ip\_1hour\_account\_cnt | IP1小时内关联账户数 | 是 |
| v\_ip\_1day\_account\_cnt | IP1天内关联账户数 | 是 |
| v\_ip\_1min\_dev\_cnt | IP1分钟内关联设备数 | 是 |
| v\_ip\_10min\_dev\_cnt | IP10分钟内关联设备数 | 是 |
| v\_ip\_1hour\_dev\_cnt | IP1小时内关联设备数 | 是 |
| v\_ip\_1day\_dev\_cnt | IP1天内关联设备数 | 是 |
| v\_account\_1min\_cnt | 账户1分钟内出现次数 | 是 |
| v\_account\_10min\_cnt | 账户10分钟内出现次数 | 是 |
| v\_account\_1hour\_cnt | 账户1小时内出现次数 | 是 |
| v\_account\_1day\_cnt | 账户1天内出现次数 | 是 |
| v\_account\_1min\_dev\_cnt | 账户1分钟内关联设备数 | 是 |
| v\_account\_10min\_dev\_cnt | 账户10分钟内关联设备数 | 是 |
| v\_account\_1hour\_dev\_cnt | 账户1小时内关联设备数 | 是 |
| v\_account\_1day\_dev\_cnt | 账户1天内关联设备数 | 是 |
| v\_account\_1min\_ip\_cnt | 账户1分钟内关联ip数 | 是 |
| v\_account\_10min\_ip\_cnt | 账户10分钟内关联ip数 | 是 |
| v\_account\_1hour\_ip\_cnt | 账户1小时内关联ip数 | 是 |
| v\_account\_1day\_ip\_cnt | 账户1天内关联ip数 | 是 |
| h\_ip\_risk\_label\_num | ip命中风险标签的个数 | 否 |
| h\_ip\_wool\_occur\_days | ip发生薅羊毛的天数 | 否 |
| h\_ip\_signup\_occur\_days | ip发生垃圾注册的天数 | 否 |
| h\_ip\_scapling\_occur\_days | ip发生黄牛的天数 | 否 |
| h\_ip\_click\_occur\_days | ip发生虚假点击的天数 | 否 |
| h\_ip\_spam\_occur\_days | ip发生垃圾信息的天数 | 否 |
| h\_ip\_sms\_occur\_days | ip发生短信轰炸的天数 | 否 |
| h\_ip\_isproxy\_http | ip是否是http代理 | 否 |
| h\_ip\_isproxy\_socks | ip是否是socks代理 | 否 |
| h\_ip\_isproxy\_vpn | ip是否是vpn代理 | 否 |
| h\_ip\_third\_risk\_cnt | ip命中第三方情报种类 | 否 |
| h\_ip\_type\_idc | ip类型是否是机房 | 否 |
| h\_ip\_isforeign | 是否是国外ip | 否 |
| h\_ip\_isreserve | 是否是保留ip | 否 |
| h\_ip\_format\_abnormal | ip格式是否异常 | 否 |
| h\_dev\_no\_id | 设备缺失 | 否 |
| h\_sign\_time | 注册时长 | 否 |
| h\_is\_sensetive\_time | 是否是敏感时段注册 | 是 |
| h\_is\_same\_addr\_mob\_ip | ip和手机号注册地址是否一致 | 否 |
| h\_ip\_10min\_cnt\_max | ip10分钟最大注册次数 | 是 |
| h\_ip\_10min\_cnt\_mean | ip10分钟平均次数 | 是 |
| h\_ip\_10min\_account\_max | ip10分钟最大注册账户 | 是 |
| h\_ip\_10min\_account\_mean | ip10分钟平均注册账户数 | 是 |
| h\_dev\_10min\_cnt\_max | 设备10分钟最大注册次数 | 是 |
| h\_dev\_10min\_cnt\_mean | 设备10分钟平均注册次数 | 是 |
| h\_dev\_10min\_account\_max | 设备10分钟最大注册账户数 | 是 |
| h\_dev\_10min\_account\_mean | 设备10分钟平均注册账户数 | 是 |
| h\_dev\_account\_cnt | 设备注册账户数 | 否 |
| h\_mob\_isdata | 是否流量卡 | 否 |

四、有监督模型构建与效果

根据客户打标数据和复杂网络关联结果，采用有监督模型分别构建模型并预测结果。通过客户打标数据，可以准确的预测其他测试数据；通过复杂网络数据，可以补充客户未识别的风险数据。具体的数据准备如下：

1、客户打标数据：使用客户标记好的黑样本85861个，白样本12443个，将黑白样本混合后，随机按照1：1划分训练集和验证集，将数据划分为web端数据和app端数据分别处理，其中app端样本数据去除掉注册时长（h\_sign\_time），其余特征一致，通过模型学习分别训练app端和web端的模型分类器，然后将两个分类器预测判断验证，将预测结果与实际标签结果进行比对，检验模型的效果，并最终对剩余未打标数据进行预测。

2、复杂网络数据：复杂网络训练模型识别出的黑样本21909条，其中与客户白样本重合18条，黑样本重合1437条。将客户白样本的12443条数据中去除复杂网络命中的18条黑样本，剩余12425条数据当做白样本与复杂网络生成的21909条黑样本进行混合，生成预测模型，并对结果进行预测。

4.1、随机森林

**4.1.1、模型思路：**是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类（对于分类算法），然后看看哪一类被选择最多，就预测这个样本为那一类

**4.1.2、模型训练及验证：**

**4.1.2.1 使用客户打标数据作为Y标签，进行训练和预测**

1. WEB端数据共15075条，训练集和验证集1：1，训练集数据7537条，验证集数据7538条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 6856 | 135 |
| 预测白 | 171 | 376 |

准确率 95.9406%，召回率 97.5665%

1. APP端数据共83229条，按照训练集和验证集1：1划分，训练集数据41614条，验证集数据41615条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 34102 | 1801 |
| 预测白 | 1820 | 3892 |

准确率 91.2988%，召回率 94.9335%

1. APP端和WEB端合并共98304条，按照训练集和验证集1：1划分，训练集数据49152条，验证集数据49152条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 41071 | 2059 |
| 预测白 | 1889 | 4133 |

准确率 91.9678%，召回率 95.6029%

1. 数据预测：

对剩余210按照随机森林预测结果的不同概率将数据的风险结果进行了不同程度的划分。

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 有风险数据 | 无风险数据 |
| 0.5 | 1769533 | 354221 |
| 0.7 | 1678557 | 445197 |
| 0.8 | 1644384 | 479370 |
| 0.9 | 1611082 | 512672 |

**4.1.2.2使用复杂网络产生的数据作为Y标签，进行训练、验证、预测：**

1. 通过复杂网络训练数据生成随机森林预测模型，然后对客户打标好的黑白样本数据进行预测，随机森林的预测概率大于0.8得出以下结果

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 1442 | 18 |
| 预测白 | 84419 | 12425 |

客户打标的18条白样本，复杂网络预测出为黑，也就是说复杂网络补充了18条黑样本。

1. 数据预测：根据不同的概率阈值，得出结果如下：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 有风险数据 | 无风险数据 |
| 0.5 | 27077 | 2096677 |
| 0.7 | 22363 | 2101391 |
| 0.8 | 21009 | 2102745 |
| 0.9 | 20725 | 2103029 |

* + 1. **结果分析**

随机森林预测概率均按照0.8计算，非常用客户打标数据、复杂网络数据预测出有风险的数据如下：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 客户标签计算出的风险数据 | 复杂网络标签计算出的风险数据 |
| 0.8 | 1644384 | 21009 |

两类数据的重合数据仅有19093条，也就是说复杂网络另外贡献了1916条新的可能风险样本

4.2、GBDT

**4.2.1、模型思路：**采用cart分类器，迭代决策树,是损失函数最小，并将所有树的结论累加。

**4.2.2、模型训练及验证：**

4.2.2.1、**使用客户打标数据作为Y标签，进行训练和预测**

1. web端数据共15075条，训练集和验证集以1：1进行划分，因此训练集数据7537条，验证集数据7538条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 6851 | 94 |
| 预测白 | 163 | 430 |

计算得出准确率96.59%,召回率97.68%。

1. APP端数据共83229条，按照训练集和验证集1：1划分，训练集数据41614条，验证集数据41615条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 34324 | 1788 |
| 预测白 | 1636 | 3867 |

计算得出准确率91.77%，召回率95.45%

1. APP端和PC端合并共98304条，按照训练集和验证集1：1划分，训练集数据49152条，验证集数据49152条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 41175 | 1882 |
| 预测白 | 1799 | 4297 |

计算得出准确率 92.51%，召回率 95.81%

1. 数据预测：

按照gbdt训练模型，对剩余数据2123754条进行预测，得出结果如下：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 有风险数据 | 占比 |
| 0.5 | 1779717 | 83.8% |
| 0.7 | 1672992 | 78.77% |
| 0.8 | 1656625 | 78.0% |
| 0.9 | 1632901 | 76.88% |

4.2.2.2**使用复杂网络产生的数据作为Y标签，进行训练、验证、预测：**

1. 模型训练及验证：

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 1438 | 5 |
| 预测白 | 84429 | 12438 |

通过复杂网络识别欺诈数据，增益了5条记录。

1. 数据预测：以0.5（默认）为预测概率阈值，识别出欺诈风险数据21467条，白样本2102288条。

**4.2.3 结果分析**

用客户打标数据、复杂网络数据预测结果，如下:

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 客户标签计算出的风险数据 | 复杂网络标签计算出的风险数据 |
| 0.5 | 1761876 | 21467 |

两类数据对比，得出重合数据有21110条，也就是说复杂网络另外贡献了357条新的可能风险样本。

4.3、XGBOOST

**4.3.1、模型设计：针对二分类问题，采用逻辑回归作为基本学习器，以logloss来衡量模型效果，通过迭代使损失函数最小得出预测的概率值。相比于gbdt，xgboost加入了对于模型复杂度的控制以及后期的剪枝处理，使得学习出来的模型更加不容易过拟合。**

**4.3.2、模型训练及验证：**

**4.3.2.1使用客户打标数据作为Y标签，进行训练和预测**

1. web端数据共15075条，训练集和验证集以1：1进行划分，因此训练集数据7537条，验证集数据7538条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 6846 | 107 |
| 预测白 | 150 | 417 |

得出召回率97.8614%，准确率98.4651%。

1. APP端数据共83229条，按照训练集和验证集1：1划分，训练集数据41614条，验证集数据41615条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 34145 | 1578 |
| 预测白 | 1815 | 4077 |

得出召回率94.9527%，准确率95.5827%。

1. APP端和WEB端合并共98304条，按照训练集和验证集1：1划分，训练集数据49152条，验证集数据49152条，其中

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 41452 | 2218 |
| 预测白 | 1509 | 3973 |

得出召回率96.4875%，准确率94.9210%

1. 数据预测：

使用xgboost模型，对剩余数据进行预测，得出有欺诈行为的数据1761876条，无风险数据361879条。

|  |  |  |
| --- | --- | --- |
| 渠道 | 有风险数据 | 无风险数据 |
| Web | 251356 | 61115 |
| app | 1510520 | 300764 |
| 总计 | 1761876 | 361879 |

**4.3.2.2使用复杂网络产生的数据作为Y标签，进行训练、验证、预测:**

* 将复杂网络的训练数据生成xgboost预测模型，然后对客户打标好的黑白样本数据进行预测，xgboost的预测概率阈值0.5，得出以下结果：

|  |  |  |
| --- | --- | --- |
|  | 实际黑 | 实际白 |
| 预测黑 | 1449 | 18 |
| 预测白 | 96855 | 98286 |

通过该模型的学习，多识别出18条客户未识别出的有风险的记录。

* 数据预测：使用该模型预测出黑样本20967，白样本2102788。
  + 1. **结果分析**

鉴于xgboost模型，使用复杂网络数据预测出有风险的数据如下：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 客户标签计算出的风险数据 | 复杂网络标签计算出的风险数据 |
| 0.5 | 1761876 | 20967 |

两类数据的重合数据有20884条，也就是说复杂网络另外贡献了83条新的可能风险样本。

**4.4三个模型结果对比**

将随机森林,gbdt,xgboost三个模型的结果进行比对，GBDT取概率阈值为0.8，随机森林取概率阈值为0.8，xgboost去概率阈值为0.5，预测结果识别为风险数据比较得出：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 渠道 | 随机森林 | GBDT | XGBOOST | 随机森林&GBDT | 随机森林预&XGBOOST | GBDT & XGBOOST | 随机森林& GBDT& XGBOOST |
| Web | 241777 | 237335 | 251355 | 233061 | 238676 | 237266 | 233047 |
| APP | 1402607 | 1419290 | 1510520 | 1390842 | 1399727 | 1416186 | 1390376 |
| 总数 | 1644384 | 1656625 | 1761875 | 1623903 | 1638403 | 1653452 | 1623423 |

可以看出，是三个模型的判别结果的重合率高达90%以上。

**五、组合模型**

采用组合模型能提高预测的精度，因此我们将随机森林、gbdt、xgboost预测结果进行组合，将概率预测值进行加权融合，权重分别设置为0.3,0.3,0.4，得出预测结果：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 有风险数据 | 无风险数据 |
| 0.5 | 1769194 | 354560 |
| 0.7 | 1668708 | 455046 |
| 0.8 | 1645327 | 478427 |
| 0.9 | 1612643 | 511111 |

关于概率阈值的选择，需要根据实际应用场景对模型的结果进行组合。比如在对拒绝准确率需求高的场景，将阈值设高一些，保证拒绝的准确性，而对召回率需求高的场景，可以增对阈值设的小一些。

复杂网络：

预测出结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 概率阈值 | 有风险数据 | 无风险数据 | 风险比例 |
| 0.5 | 21404 | 2102350 | 1.01% |
| 0.7 | 20967 | 2102787 | 0.99% |
| 0.8 | 20962 | 2102792 | 0.99% |
| 0.9 | 20933 | 2102821 | 0.99% |

表8 复杂网络样本预测结果表

通过复杂网络特征，识别出有风险的数据在1%左右。此部分数据可以对预测数据做一个补充。

### 客户样本和复杂网络样本对比

将复杂网络预测结果和客户预测结果进行对比，可以看出增益数据如下：

|  |  |  |
| --- | --- | --- |
| 概率阈值 | 重合 | 增益数据 |
| 0.5 | 21331 | 73 |
| 0.7 | 20324 | 643 |
| 0.8 | 19201 | 1761 |
| 0.9 | 18076 | 2857 |

表9 预测结果对比表

从中可见，通过复杂网络数据训练的模型，同客户样本训练模型相比，有一定的增益，如概率阈值在0.8时，增益数为1761。概率值越高，识别出有风险的数据越准确，复杂网络贡献的增益数也越高，但会有一部分的漏抓。

复杂网络数据比对

复杂网路黑样本25940。

将复杂网络团伙数>=5作为黑样本，与客户打标黑样本重合1550条，和白样本重合31条。

预测出风险标签24359，白样本2099396

复杂网络这次跑出来没有增益。